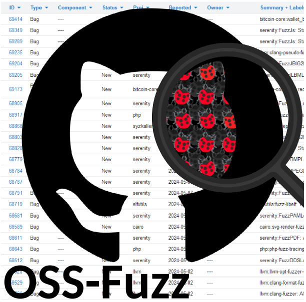
Quantitative Robustness for Vulnerability Assessment

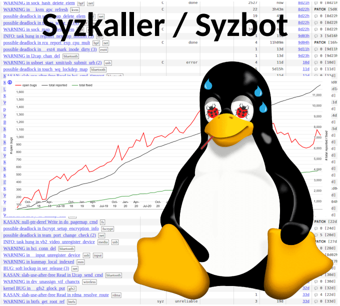
Guillaume Girol - Guilhem Lacombe - Sébastien Bardin

PLDI COPENHAGEN 2024

Bug-finding techniques are really good at finding bugs!

Fuzzing





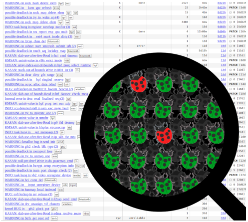
Other successful techniques

- symbolic execution (Klee, Angr, Binsec)
- abstract interpretation (Frama-C, Infer)
- ...

Not all bugs are created equal

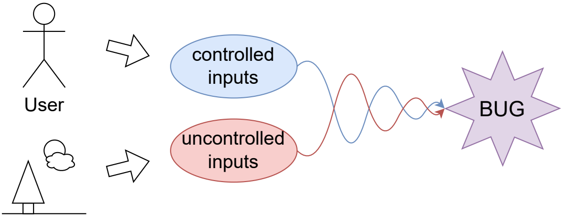
Bug impacts

infinite loop, memory corruption...



Bug reproducibility

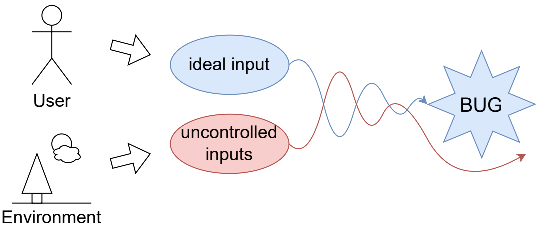
dependency on uncontrolled inputs
⇒ randomness, stack canaries, scheduling, undefined behaviour, uninitialized memory...



Evaluating bug reproducibility with robust reachability

Girol et al., CAV 2021, FMSD 2022

> **Robust reachability**
>
> ∃ a *controlled input* triggering the bug ∀ *uncontrolled input*



in the real-world: CVE-2019-20839, CVE-2019-15900, CVE-2019-19307...

What about mostly-robust bugs?

Two different bugs

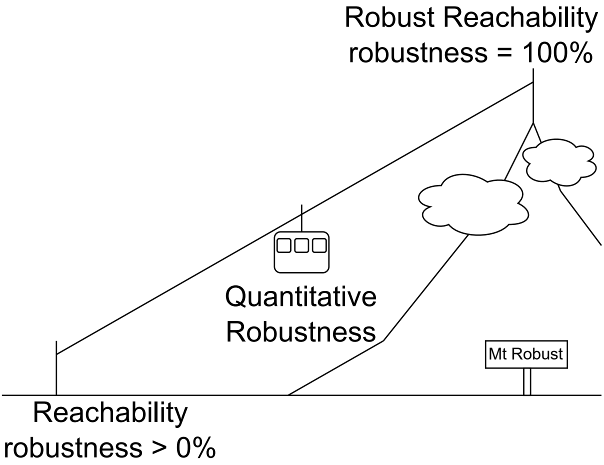
```
int controlled = INPUT;
int uncontrolled = NONDET;
...
if(uncontrolled - controlled == 1)
    //bug 1
...
if(uncontrolled & controlled == 1)
    //bug 2
```

- **bug 1:** extremely unlikely ($\frac{1}{2^{31}}$)
- **bug 2:** very likely with *controlled* $= 1$ ($\frac{1}{2}$)

- both are reachable

- none are robustly reachable

We need a quantitative measure of robustness



Goals and challenges

Goals

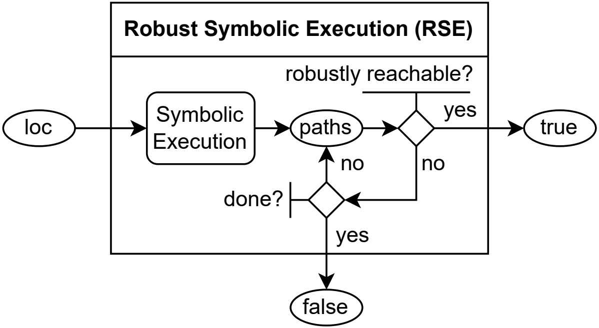
- formally define quantitative robustness
- design algorithms to measure it
- automation + scalability

Challenges

- scalability of quantitative analysis (ex: model counting)
- improve performance over robust symbolic execution (RSE)

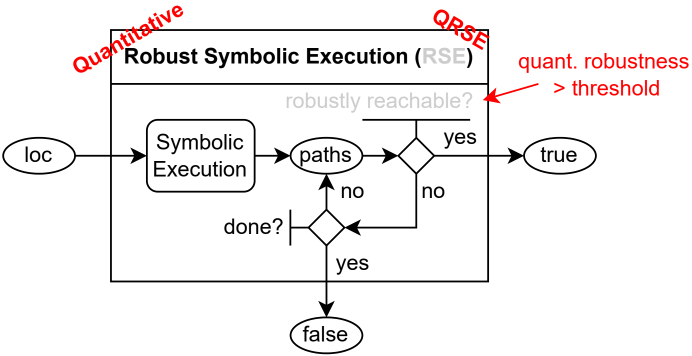
Our approach

robust reachability



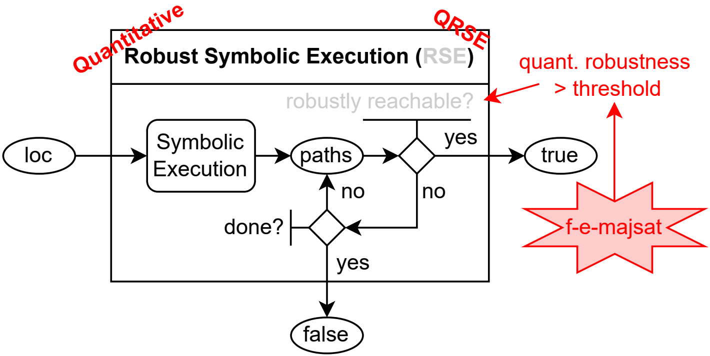
Our approach

robust reachability ⇒ quantitative robustness



Our approach

robust reachability ⇒ quantitative robustness



f-e-majsat: counting + optimization problem related to AI
∼ *unknown in security*

Contributions

Quantitative robustness

formal definition + theorems

QRSE

- quantitative version of RSE
- path-wise quantitative robustness reduced to *f-e-majsat*
  $\Rightarrow$ security application of *f-e-majsat*
- *Relax*, a new approximate *f-e-majsat* solving algorithm

Implementation

- **BINSEC/QRSE:** binary-level QRSE
- **Popcon:** front-end for f-e-majsat solvers (bitvectors)
- experiments with realistic security-related case studies

Defining Quantitative robustness

Threat model (Girol et al., CAV 2021)

program $\mathcal{P}$, targeted location *loc*

- **controlled inputs** $\in \mathcal{A}$ *countable*
- **uncontrolled inputs** $\in \mathcal{X}$ *countable*, uniformly distributed

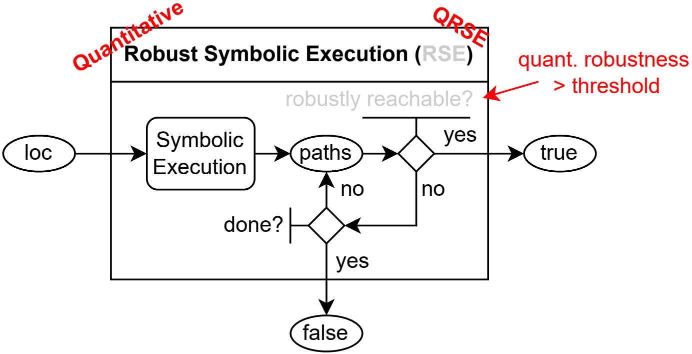
Quantitative robustness (finite input space)

max proportion of x reaching *loc* for a fixed a

$$q_{loc} \triangleq \frac{max_{a\in\mathcal{A}}|\{x\in\mathcal{X} \text{ s.t. } \mathcal{P}(a,x) \text{ reaches } loc\}|}{|\mathcal{X}|}$$

(read the paper for the infinite input space definition)

From RSE to QRSE

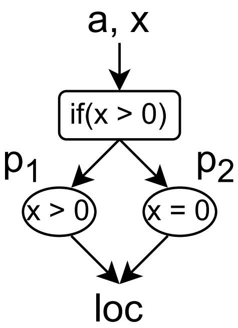


- **QRSE+:** QRSE with path merging
- correctness: ✓
- k-completeness (path lengths $\leq k$): QRSE+ ✓

Path merging and deduction power

Path merging

- multi-path constraint
- $p_1 \vee p_2 \sim true$
- added complexity $\Rightarrow$ **scalability issues**

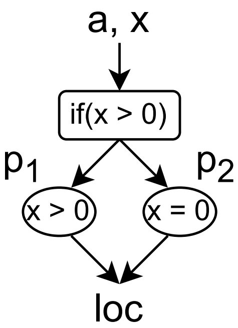


Path merging and deduction power

Path merging

- multi-path constraint
- $p_1 \vee p_2 \sim true$
- added complexity $\Rightarrow$ **scalability issues**

- **SE:** deduction power $=$ $reach(p_1), reach(p_2), reach(p_1 \vee p_2)$
- **RSE:** deduction power $\nearrow\nearrow\nearrow$ $\neg RSE(p_1), \neg RSE(p_2), RSE(p_1 \vee p_2)$
- **QRSE:** deduction power $\nearrow$ $QRSE(p_1) > 99\% \Rightarrow q_{loc} > 99\%$ $+$ guarantees on robustness lower bound after branches



What quantitative robustness is *not*

context: binary analysis ⇒ bitvectors ⇒ finite input space

Model counting (#sat)

- **application:** probabilistic reachability
- **here:** # reaching inputs
- **issue:** controlled inputs are not random

Projected model counting

- **application:** quantitative information flow, channel capacity...
- **here:** # reaching uncontrolled inputs (any controlled inputs)
- **issue:** choice of best controlled input

(also not weighted maxSMT)

What quantitative robustness *is*

f-e-majsat (Littman et al., JAIR 1998)

f: propositional formula

$$femajsat_{\mathcal{A}}(f) \triangleq max_{a \in \mathcal{A}} \#(f|_a)$$

known applications: probabilistic planning, Bayesian networks...

What quantitative robustness *is*

f-e-majsat (Littman et al., JAIR 1998)

f: propositional formula

$$femajsat_{\mathcal{A}}(f) \triangleq max_{a \in \mathcal{A}} \#(f|_a)$$

known applications: probabilistic planning, Bayesian networks...

Existing algorithms

Algorithm	Author(s)	Conference
DC-SSAT	Majercik et al.	AAAI 2005
Constrained	Huang	ICAPS 2006
Complan	Huang	ICAPS 2006
Complan+	Pipatsrisawat et al.	IJCAI 2009
MaxCount	Fremont et al.	AAAI 2017
SsatABC	Lee et al.	IJCAI 2018

$\Rightarrow$ untested on quantitative robustness (-like) instances

Efficient approximation of *f-e-majsat*

Basic exact approach

- compile constraints to decision-DNNF form
- **additional constraint:** $(\mathcal{A}, \mathcal{X})$-layering
- model counting in linear time (Darwiche, 2001)

issue: compilation is *hard* ($|\mathcal{X}| \nearrow \Rightarrow$ speed $\searrow$)

Our *Relax* algorithm

- **relaxation:** $(\mathcal{A} \cup \mathcal{R}, \mathcal{X} \backslash \mathcal{R})$-layering
- interval
- $Relax_{-}(f) \leq femajsat_{\mathcal{A}}(f) \leq Relax_{+}(f) \leq 2^{|\mathcal{R}|}\ Relax_{-}(f)$

Introduction

Quantitative Robustness and QRSE

Reduction to *f-e-majsat*

Experimental Results

Conclusion

Bonus

Research questions

- Is QRSE more precise than RSE in practice?
- Can we avoid path merging?
- What are the best f-e-majsat solvers for quantitative robustness?

Is QRSE more precise than RSE in practice?

RSE benchmark ($> 20\%$?)

Method	OK	FN	T
RSE	37	9	2
RSE+	40	6	2
QRSE	47	0	1
QRSE+	46	0	2

Fault analysis benchmark

q_{loc}	SE	RSE	QRSE
100%	-	0/0	0/0
$> 20\%$	-	-	2/2
$[10^{-6}; 20\%]$	-	-	10/10
$< 10^{-6}$	-	-	27/27
$> 0\%$	39/39	-	39/39

- no more false negatives!
- distinguishes *nearly robust* from *nearly unreproducible*

Can we avoid path merging?

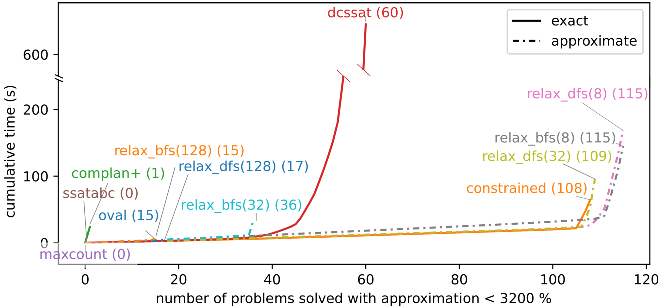
Case study: stack buffer overflow in libvncserver (CVE-2019-20839)

canary	SE	RSE	RSE+	QRSE	QRSE+
no	$> 0\%$	✗	✓	1 path $> 20\%$ ✓	$> 20\%$ ✓
yes	$> 0\%$	⏱ ✗	⏱ ✗	all paths $< 20\%$ ✓	⏱ ✗

$\Rightarrow$ useful results without path merging!

What are the best f-e-majsat solvers for quantitative robustness?

Benchmark with 117 formulas from previous experiments



- **surprise 1:** almost all existing algorithms perform poorly ☹
- **surprise 2:** naive algorithm *Constrained* ☺ (108/117)
- *Relax* with $|\mathcal{R}| = 8$ ☺ (115/117)

Conclusion

- Bug replicability is important!
- Quantitative robustness $\Rightarrow$ precise indicator of replicability
- Measured with QRSE, reduction to *f-e-majsat*
- best algorithms: *Constrained* and *Relax*
- BINSEC/QRSE

Conclusion

- Bug replicability is important!
- Quantitative robustness ⇒ precise indicator of replicability
- Measured with QRSE, reduction to *f-e-majsat*
- best algorithms: *Constrained* and *Relax*
- BINSEC/QRSE

Possible improvements

concept to handle	Definitions	SE	Solver
non-uniform input distrib.	easy	hard	hard
hyper-safety properties	easy	easy	easy
hyperproperties, liveness	hard	hard	hard
string & LIA theories	ok	medium	hard
dense input spaces	easy	hard	hard

The end

Thank you for your attention.

Any questions?

k-completeness of QRSE+

k-completeness

$P|^{\leq k}$ = restriction of P to traces of length $\leq k$
k-complete for $P \iff$ complete for $P|^{\leq k}$

- 0 or 1 path of length $\leq k$ per input
  $\Rightarrow$ finite number of finite paths
- QRSE+ can explore and merge them all
  $\Rightarrow$ constraint for reaching *loc* in $P|^{\leq k} \subset$ final constraint
- $q_{P|^{\leq k},loc} \geq Q \Rightarrow \text{QRSE+}(P, loc) \geq Q$
  (assuming no timeouts or errors)

Branching theorem

Quantitative robustness pseudo-conservation

$p_1, \ldots, p_n$ paths in P, $P|^{p_1,\ldots,p_n}$ = restriction of P to $p_1, \ldots, p_n$

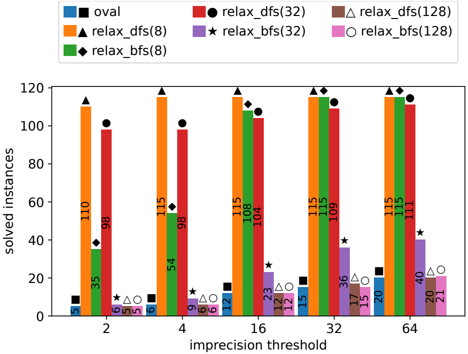
$$\exists\ i \text{ s.t. } q_{P|^{p_i},loc} \geq \frac{1}{n} q_{P|^{p_1,\ldots,p_n},loc}$$

idea of the proof:

- show $q_{P|^{p,p'},loc} \leq q_{P|^{p},loc} + q_{P|^{p'},loc}$
- contradiction: $q_{P|^{p_i},loc} < \frac{1}{n} q_{P|^{p_1,\ldots,p_n},loc}\ \forall\ i$
  $\Rightarrow q_{P|^{p_1,\ldots,p_n},loc} < n \times \frac{1}{n} q_{P|^{p_1,\ldots,p_n},loc}$

Practical precision of approximate solvers

Solved problems function of imprecision threshold



- *Relax*: $|\mathcal{R}| = 8$ with 4x imprecision ⇒ ☺
- better than theoretical bounds!